

# Package ‘freestiler’

April 21, 2026

**Title** Create Vector Tiles from Spatial Data

**Version** 0.1.6

**Date** 2026-04-13

**Description** Create vector tile archives in 'PMTiles' format from 'sf' spatial data frames. Supports 'Mapbox Vector Tile' ('MVT') and 'MapLibre Tile' ('MLT') output formats. Uses a 'Rust' backend via 'extendr' for fast, in-memory tiling with zero external system dependencies.

**License** MIT + file LICENSE

**URL** <https://walker-data.com/freestiler/>,  
<https://github.com/walkerke/freestiler/>

**BugReports** <https://github.com/walkerke/freestiler/issues>

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Imports** sf

**Suggests** arrow, DBI, duckdb, httpuv, jsonlite, mapgl, testthat (>= 3.0.0), withr

**Config/testthat/edition** 3

**Config/rextendr/version** 0.4.2.9000

**SystemRequirements** Cargo (Rust's package manager), rustc >= 1.70

**Depends** R (>= 4.2)

**NeedsCompilation** yes

**Author** Kyle Walker [aut, cre]

**Maintainer** Kyle Walker <kyle@walker-data.com>

**Repository** CRAN

**Date/Publication** 2026-04-21 10:52:20 UTC

## Contents

freestile	2
freestile_file	4
freestile_layer	6
freestile_query	7
serve_tiles	9
stop_server	10
view_tiles	11
<b>Index</b>	<b>13</b>

---

freestile	<i>Create vector tiles from spatial data</i>
-----------	--

---

## Description

Creates a PMTiles archive containing vector tiles from one or more sf data frames. Supports both Mapbox Vector Tile (MVT) and MapLibre Tile (MLT) formats, multi-layer output, feature dropping, point clustering, and feature coalescing.

## Usage

```
freestile(
  input,
  output,
  layer_name = NULL,
  tile_format = "mvt",
  min_zoom = 0L,
  max_zoom = 14L,
  base_zoom = NULL,
  drop_rate = NULL,
  cluster_distance = NULL,
  cluster_maxzoom = NULL,
  coalesce = FALSE,
  simplification = TRUE,
  generate_ids = TRUE,
  overwrite = TRUE,
  quiet = FALSE
)
```

## Arguments

input	An sf data frame, or a named list of sf/freestile_layer objects for multi-layer output.
output	Character. Path for the output .pmtiles file.
layer_name	Character. Name for the tile layer. If NULL, derived from the output filename. Only used for single-layer input.

<code>tile_format</code>	Character. Tile encoding format: "mvt" (default) for Mapbox Vector Tiles or "mlt" for MapLibre Tiles.
<code>min_zoom</code>	Integer. Minimum zoom level (default 0).
<code>max_zoom</code>	Integer. Maximum zoom level (default 14).
<code>base_zoom</code>	Integer. Zoom level at and above which all features are present (no dropping). NULL (default) uses each layer's own <code>max_zoom</code> . The drop-rate curve is also computed relative to <code>base_zoom</code> , so lowering it produces gentler thinning at low zooms. Inspired by tippecanoe's <code>-B / --base-zoom</code> .
<code>drop_rate</code>	Numeric. Exponential drop rate for feature thinning (e.g. 2.5). At each zoom level below <code>base_zoom</code> , features are retained at a rate of $1/\text{drop\_rate}^{(\text{base\_zoom} - \text{zoom})}$ . Points are thinned using spatial ordering; polygons/lines are thinned by area. NULL (default) disables drop-rate thinning.
<code>cluster_distance</code>	Numeric. Pixel distance for point clustering. Points within this radius are merged into cluster features with a <code>point_count</code> attribute. NULL (default) disables clustering.
<code>cluster_maxzoom</code>	Integer. Maximum zoom level for clustering. Above this zoom, individual points are shown. Default is <code>max_zoom - 1</code> .
<code>coalesce</code>	Logical. Whether to merge features with identical attributes within each tile (default FALSE). Lines sharing endpoints are merged; polygons are grouped into MultiPolygons.
<code>simplification</code>	Logical. Whether to snap geometries to the tile pixel grid at each zoom level (default TRUE). This provides zoom-adaptive simplification and prevents slivers between adjacent polygons.
<code>generate_ids</code>	Logical. Whether to assign sequential feature IDs (default TRUE).
<code>overwrite</code>	Logical. Whether to overwrite existing output file (default TRUE).
<code>quiet</code>	Logical. Whether to suppress progress messages (default FALSE).

### Details

Input data in any coordinate reference system (CRS) is automatically reprojected to WGS84 (EPSG:4326) before tiling.

### Value

The output file path (invisibly).

### Examples

```
## Not run:
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))

# Single layer
freestile(nc, "nc.pmtiles", layer_name = "counties")
```

```
# Multi-layer
pts <- st_centroid(nc)
freestyle(
  list(counties = nc, centroids = pts),
  "nc_layers.pmtiles"
)

# With dropping and coalescing
freestyle(nc, "nc_drop.pmtiles", drop_rate = 2.5, coalesce = TRUE)

# With point clustering
freestyle(pts, "pts.pmtiles", cluster_distance = 50, cluster_maxzoom = 8)

## End(Not run)
```

---

freestyle_file	<i>Create vector tiles from a spatial file</i>
----------------	--

---

## Description

Reads a GeoParquet, GeoPackage, Shapefile, or other spatial file directly into the tiling engine. Input data in any coordinate reference system is automatically reprojected to WGS84 (EPSG:4326) before tiling.

## Usage

```
freestyle_file(
  input,
  output,
  layer_name = NULL,
  tile_format = "mvt",
  min_zoom = 0L,
  max_zoom = 14L,
  base_zoom = NULL,
  drop_rate = NULL,
  cluster_distance = NULL,
  cluster_maxzoom = NULL,
  coalesce = FALSE,
  simplification = TRUE,
  overwrite = TRUE,
  quiet = FALSE,
  engine = "geoparquet"
)
```

**Arguments**

input	Character. Path to the input spatial file.
output	Character. Path for the output .pmtiles file.
layer_name	Character. Name for the tile layer. If NULL, derived from the output filename.
tile_format	Character. "mvt" (default) or "mlt".
min_zoom	Integer. Minimum zoom level (default 0).
max_zoom	Integer. Maximum zoom level (default 14).
base_zoom	Integer. Zoom level at and above which all features are present. NULL (default) uses max_zoom.
drop_rate	Numeric. Exponential drop rate. NULL (default) disables.
cluster_distance	Numeric. Pixel distance for clustering. NULL disables.
cluster_maxzoom	Integer. Max zoom for clustering. Default max_zoom - 1.
coalesce	Logical. Whether to merge features with identical attributes (default FALSE).
simplification	Logical. Whether to snap geometries to the tile pixel grid (default TRUE).
overwrite	Logical. Whether to overwrite existing output (default TRUE).
quiet	Logical. Whether to suppress progress (default FALSE).
engine	Character. Backend engine: "geoparquet" (default, for GeoParquet files) or "duckdb" (for any file format DuckDB supports).

**Details**

The GeoParquet engine requires compilation with `FREESTILER_GEOPARQUET=true`. The DuckDB engine uses the Rust DuckDB backend when included in the build (enabled by default for native builds), or falls back to the R duckdb package. Control backend selection with `options(freestiler.duckdb_backend = "auto"|"rust"|"r")`.

**Value**

The output file path (invisibly).

**Examples**

```
## Not run:
freestile_file("data.parquet", "output.pmtiles")
freestile_file("data.gpkg", "output.pmtiles", engine = "duckdb")

## End(Not run)
```

---

freestile_layer	<i>Create a layer specification with per-layer zoom range</i>
-----------------	---

---

### Description

Wraps an sf object with optional per-layer zoom range overrides for use in multi-layer tile generation.

### Usage

```
freestile_layer(input, min_zoom = NULL, max_zoom = NULL)
```

### Arguments

input	An sf data frame.
min_zoom	Integer. Minimum zoom level for this layer. If NULL, uses the global min_zoom from freestile().
max_zoom	Integer. Maximum zoom level for this layer. If NULL, uses the global max_zoom from freestile().

### Value

A freestile\_layer object (list with class attribute).

### Examples

```
## Not run:
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
roads <- st_read("roads.shp")

freestile(
  list(
    counties = freestile_layer(nc, min_zoom = 0, max_zoom = 10),
    roads = freestile_layer(roads, min_zoom = 8, max_zoom = 14)
  ),
  "layers.pmtiles"
)

## End(Not run)
```

---

freestyle_query	<i>Create vector tiles from a DuckDB SQL query</i>
-----------------	--

---

## Description

Executes a SQL query via DuckDB's spatial extension and pipes the results into the tiling engine. Uses the Rust DuckDB backend when included in the build (enabled by default for native builds), or falls back to the R duckdb package. Control backend selection with `options(freestiler.duckdb_backend = "auto"|"rust"|"r")`.

## Usage

```
freestyle_query(
  query,
  output,
  db_path = NULL,
  layer_name = NULL,
  tile_format = "mvt",
  min_zoom = 0L,
  max_zoom = 14L,
  base_zoom = NULL,
  drop_rate = NULL,
  cluster_distance = NULL,
  cluster_maxzoom = NULL,
  coalesce = FALSE,
  simplification = TRUE,
  overwrite = TRUE,
  quiet = FALSE,
  source_crs = NULL,
  streaming = "auto"
)
```

## Arguments

query	Character. A SQL query that returns a geometry column. DuckDB spatial functions like <code>ST_Read()</code> and <code>read_parquet()</code> are available. Multi-statement SQL is supported: setup statements (e.g., <code>LOAD h3;</code> ) are executed first, then the final <code>SELECT</code> is used for tiling.
output	Character. Path for the output <code>.pmtiles</code> file.
db_path	Character. Path to a DuckDB database file, or <code>NULL</code> (default) for an in-memory database.
layer_name	Character. Name for the tile layer. If <code>NULL</code> , derived from the output filename.
tile_format	Character. <code>"mvt"</code> (default) or <code>"mlt"</code> .
min_zoom	Integer. Minimum zoom level (default 0).
max_zoom	Integer. Maximum zoom level (default 14).

base_zoom	Integer. Zoom level at and above which all features are present. NULL (default) uses max_zoom.
drop_rate	Numeric. Exponential drop rate. NULL (default) disables.
cluster_distance	Numeric. Pixel distance for clustering. NULL disables.
cluster_maxzoom	Integer. Max zoom for clustering. Default max_zoom - 1.
coalesce	Logical. Whether to merge features with identical attributes (default FALSE).
simplification	Logical. Whether to snap geometries to the tile pixel grid (default TRUE).
overwrite	Logical. Whether to overwrite existing output (default TRUE).
quiet	Logical. Whether to suppress progress (default FALSE).
source_crs	Character or NULL. CRS of the geometry returned by query, for example "EPSG:4326" or "EPSG:4267". Used only by the R duckdb fallback; ignored by the Rust DuckDB backend.
streaming	Character. DuckDB query execution mode: "auto" (default) enables the streaming point pipeline for large queries, "always" forces it, and "never" uses the existing in-memory path.

### Details

When using the R fallback, `source_crs` must be supplied explicitly so the query result can be interpreted or reprojected correctly. Pass "EPSG:4326" if the SQL already returns WGS84 geometry, or the source CRS string (for example "EPSG:4267") to have DuckDB reproject to WGS84 before tiling. For file-based input where the CRS is embedded in the file, use `freestyle_file()` with `engine = "duckdb"` instead, which auto-detects the source CRS.

### Value

The output file path (invisibly).

### Examples

```
## Not run:
# Query a GeoParquet file
freestyle_query(
  "SELECT * FROM read_parquet('data.parquet') WHERE pop > 50000",
  "output.pmtiles"
)

# Query a Shapefile
freestyle_query(
  "SELECT * FROM ST_Read('counties.shp')",
  "counties.pmtiles"
)

# Query with an existing DuckDB database
freestyle_query(
  "SELECT * FROM my_table WHERE region = 'West'",
```



```
"west.pmtiles",  
  db_path = "my_database.duckdb"  
)  
  
## End(Not run)
```

---

`serve_tiles`*Serve PMTiles files via local HTTP server with CORS*

---

### Description

Start a local HTTP server to serve PMTiles files with CORS headers and HTTP range request support. This allows PMTiles to be consumed by mapgl and MapLibre GL JS. The server runs in the background and can be stopped with `stop_server()`.

### Usage

```
serve_tiles(path, port = 8080)
```

### Arguments

<code>path</code>	Path to a directory containing PMTiles files, or a single PMTiles file. If a single file, its directory will be served.
<code>port</code>	Port number for the HTTP server. Default is 8080.

### Details

If a server is already running on the requested port, it is stopped first.

The server uses `httpuv` (a dependency of Shiny) to serve static files with the CORS and range-request headers that PMTiles requires. Works well for files up to ~1 GB. For larger files, consider an external server like `npx http-server /path --cors -c-1`.

### Value

Invisibly returns a list with `url`, `port`, and `dir`. The server handle is stored internally so it can be stopped with `stop_server()`.

### See Also

[stop\\_server\(\)](#), [view\\_tiles\(\)](#)

**Examples**

```
## Not run:
# Serve a directory
serve_tiles("/tmp/tiles")

# Serve a single file (its directory is served)
serve_tiles("us_bgs.pmtiles")

# Stop when done
stop_server()

## End(Not run)
```

---

stop\_server

*Stop a local tile server*

---

**Description**

Stop a local tile server

**Usage**

```
stop_server(port = NULL)
```

**Arguments**

port                   Port number to stop, or NULL to stop all running servers.

**Value**

Invisibly returns TRUE if a server was stopped.

**See Also**

[serve\\_tiles\(\)](#)

**Examples**

```
## Not run:
serve_tiles("tiles/")
stop_server()       # stop all
stop_server(8080)   # stop specific port

## End(Not run)
```

---

`view_tiles`*Quickly view a PMTiles file on an interactive map*

---

### Description

Starts a local tile server (if needed) and creates an interactive mapgl map showing the tileset. Layer type and styling are auto-detected from the PMTiles metadata when possible.

### Usage

```
view_tiles(  
  input,  
  layer = NULL,  
  layer_type = NULL,  
  color = NULL,  
  opacity = 0.5,  
  port = 8080,  
  promote_id = NULL  
)
```

### Arguments

<code>input</code>	Path to a local <code>.pmtiles</code> file.
<code>layer</code>	Character. Source layer name to display. If NULL (default), the first layer in the tileset is used.
<code>layer_type</code>	Character. Map layer type: "fill", "line", or "circle". If NULL, auto-detected from the PMTiles metadata geometry type.
<code>color</code>	Fill, line, or circle color. Default is "navy" for fill and line layers, "steelblue" for circle layers.
<code>opacity</code>	Numeric opacity (0–1). Default is 0.5.
<code>port</code>	Port for the local tile server. Default is 8080.
<code>promote_id</code>	Character. Property name to use as the feature ID for hover interactivity. If NULL, no feature promotion is used.

### Value

A mapgl map object (can be piped into further mapgl operations).

### See Also

[serve\\_tiles\(\)](#), [freestile\(\)](#)

**Examples**

```
## Not run:
freestyle(nc, "nc.pmtiles", layer_name = "counties")
view_tiles("nc.pmtiles")

# Override auto-detection
view_tiles("roads.pmtiles", layer_type = "line", color = "red")

# Point data
view_tiles("airports.pmtiles", layer_type = "circle", color = "orange")

## End(Not run)
```

# Index

freestile, [2](#)  
freestile(), [11](#)  
freestile\_file, [4](#)  
freestile\_file(), [8](#)  
freestile\_layer, [6](#)  
freestile\_query, [7](#)

serve\_tiles, [9](#)  
serve\_tiles(), [10](#), [11](#)  
stop\_server, [10](#)  
stop\_server(), [9](#)

view\_tiles, [11](#)  
view\_tiles(), [9](#)